IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BOARD OF PATENT APPEALS AND INTERFERENCES

Applicants:	Curtis HEISEY, et al.	**Docket No:**	3740.US.P
Serial Number:	10/016,597	**Group Art Unit:**	2192
Filed:	October 26, 2001	**Examiner:**	Eric KISS
Re:	Intelligent Device Upgrade Engine		

December 22, 2006

Mail Stop Appeal Brief – Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450

APPEAL BRIEF

Dear Sir:

The Applicants hereby submit the reasons for our Appeal to the Board of Patent Appeals and Interferences, filed October 12, 2006, from the last decision of the Examiner. This Appeal is in response to the final Office Action mailed October 6, 2006. This is also in response to the Notice of Panel Decision from Pre-Appeal Brief Review mailed December 11, 2006.

Payment of $500 for the filing of this Appeal Brief is included with this online filing. Applicants believe that no other fees are due at this time. However, should the commissioner determine that additional fees are required, the commissioner is authorized to charge deposit account 503650 for any fees associated herein.

TABLE OF CONTENTS

I. REAL PARTY IN INTEREST

3Com Corporation, a Delaware Corporation based at 350 Campus Drive in Marlborough Massachusetts is the real party in interest, by virtue of an assignment from the Applicants on June 11, 2001, June 25, 2001 and June 27, 2001. This assignment was recorded by the US Patent and Trademark Office on July 9, 2001 at Reel 011962 Frame 0479. A second assignment from the Applicants to 3Com Corporation, dated September 17, 2001, September 19, 2001, and October 4, 2001 was recorded by the US Patent and Trademark Office on October 26, 2001 at Reel 12384/0906.

II. RELATED APPEALS AND INTERFERENCES

This Appeal was subject to the Pre-Appeal Brief process through the filing of a Pre-Appeal Brief on October 12, 2006. The Pre-Appeal Panel issued a decision on December 11, 2006 recommending that this Appeal proceed to the Board of Patent Appeals and Interferences.

The Applicants and the Applicants' representatives know of no other appeals or interferences which will directly affect or be directly affected by or have a bearing on the Board's decision in this appeal.

III. STATUS OF CLAIMS

Claims 1-38 are pending in this application. Claims 1, 10, 18, 19, 20, and 33 are independent. Claims 1-38 stand rejected. This Appeal Brief addresses Claims 1-18 as presented in the July 6, 2006 "Response to Office Action". Claims 19-38 are not being pursued in this Appeal. A copy of the claims can be found in the Appendix of this Appeal Brief.

IV. STATUS OF AMENDMENTS

Applicants have not filed any amendments subsequent to the Final Office Action.

V. SUMMARY OF THE CLAIMED SUBJECT MATTER

A. Independent Claim 1

Independent claim 1 recites a system for replacing a code image in an embedded device (Fig 2 item 48, Fig 5, item 106). The system comprises a control program (Fig 5, item 100, paragraph [0039]) that responds to a user command received through a user interface (Fig 2 item 30, paragraph [0025]) by issuing device commands (Fig 5, item 108, paragraph [0056]) in order to replace a code image within the embedded device (paragraphs [0033], [0034], [0036], [0059]). A monitoring program (Fig 5, item 102, paragraph [0039], [0040]), operating asynchronously with respect to the control program (paragraph [0038]), generates event indications in response to detecting changes in one or more predetermined attributes within the embedded device (Abstract, paragraphs [0009], [0055]). The monitoring program (Fig 5, item 102) forwards the event indications to the control program (paragraph [0055]). The code image is replaced in response to the event indication (paragraphs [0033], [0034], [0036], [0059]). (See paragraph [0009] for an overview of the above.)

For purposes of this Appeal, dependent claims 2-9 are grouped with independent claim 1.

B. Independent Claim 10

Independent claim 10 recites a method for replacing a code image in an embedded device (Fig 2 item 48, Fig 5, item 106). The method comprises the issuance, in response to a user command (Fig 2 item 30, paragraph [0025]), of a number of device commands to an embedded device (Fig 5, item 108, paragraph [0056]) from a control program (Fig 5, item 100, paragraph [0039]) wherein one command replaces a code image within the embedded device (paragraphs [0033], [0034], [0036], [0059]). The method further comprises the asynchronous generation of an event in response to detection of changes in one or more predetermined attributes within the embedded device (Abstract, paragraphs [0009], [0055]). This

event is generated by a monitoring program (Fig 5, item 102, paragraph [0039], [0040]), and the event is forwarded to the control program (paragraph [0055]). The method also comprises the replacement of the code image in response to the event indication (paragraphs [0033], [0034], [0036], [0059]). (See paragraph [0009] for an overview of the above.)

For purposes of this Appeal, dependent claims 11-17 are grouped with independent claim 10.

C. Independent Claim 18

Independent claim 18 recites a computer program product on a computer readable medium for replacing a software image in an embedded device (Fig 2 item 48, Fig 5, item 106). The product comprises a control program (Fig 5, item 100, paragraph [0039]) that responds to a user command (Fig 2 item 30, paragraph [0025]) by issuing device commands (Fig 5, item 108, paragraph [0056]) in order to replace a code image within the embedded device (paragraphs [0033], [0034], [0036], [0059]). A monitoring program (Fig 5, item 102, paragraph [0039], [0040]), operating asynchronously with respect to the control program (paragraph [0038]), generates event indications in response to detecting changes in one or more predetermined attributes within the embedded device (Abstract, paragraphs [0009], [0055]). The monitoring program (Fig 5, item 102) forwards the event indications to the control program (paragraph [0055]). At least one device command replaces the code image in the embedded device and one device command is generated in response to the event indication (paragraphs [0033], [0034], [0036], [0059]). (See paragraph [0009] for an overview of the above.)

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Claims 1 to 37 are pending in this application. Claims 1, 10, 18, 19, 20, and 33 are independent.

Claims 1-17 and 19 stand rejected by the Examiner under 35 U.S.C. § 102(e) in view of U.S. Patent Application No. 2003/0126195, filed by Daniel A. Reynolds *et al.* on April 10, 2001 (hereinafter, "Reynolds"). Claim 18 stands rejected by the Examiner under 35 U.S.C. § 103(a) as being unpatentable over Reynolds in view of U.S. Patent No. 6,549,943, issued to Maximilian Spring *et al.* on April 15, 2003 (hereinafter, "Spring"). Claims 20-37 stand rejected by the Examiner under 35 U.S.C. § 103(a) as being unpatentable over Reynolds in view of U.S. Patent Application No. 2001/0055017, filed by Bas Ording *et al.* on January 5, 2001 (hereinafter, "Ording").

Only claims 1-18 are subject for this Appeal Brief. Claims 19-38 are not being pursued in this Appeal.

VII. ARGUMENT

A. The Claims

The application covers a system for replacing a code image in an embedded device. In the system tool, a control program responds to a user command received through a user interface by issuing device commands in order to replace a code image within the embedded device. A monitoring program, operating asynchronously with respect to the control program, generates event indications in response to detecting changes in one or more attribute within the embedded device. The monitoring program forwards the event indications to the control program.

Independent claims 1, 10, and 18 recite, either word for word or with similar language, "...monitoring program code, asynchronous with respect to said control program code, for generating at least one event indication in response to a **change** of at least one **predetermined attribute of said embedded device** and forwarding said at least one event indication to said control program code...".

This recitation requires a change of an attribute in the embedded device to be detected by a monitoring program. The attribute must be in the embedded device, and must be predetermined.

B. Reynolds

Reynolds teaches that "[a] common command interface (CCI) provides an interface abstraction allowing network device applications to maintain one set of code for each command regardless of which command interface (e.g., web, CLI, NMS, etc.) initiates the command.... The interface abstraction allows new applications including additional commands to be added to a network device and existing applications to be dynamically upgraded to include new and/or modified commands without having to modify the CCI." (Reynolds, Abstract). Within Reynolds, there are several paragraphs ([0504] through [0506]) that describe downloading firmware from a directory into an embedded device.

However, there are no teachings in Reynolds of a monitoring program doing anything in response to a change in an attribute of the embedded device. See Reynolds at paragraphs [0504] through [0506], as cited in the office action.

[**0504**] Master SMS 184 periodically polls installation directory **1222** for new sub-directories including new releases, for example, release 1.1 **1218** in sub-directory **1220**. When the master SMS detects a new release, it opens (and decompresses, if necessary) the packaging list in the new sub-directory and verifies that each software component listed in the packaging list is also stored in the new sub-directory. The master SMS then performs a checksum on each software component and compares the generated checksum to the checksum appended to the software component.

[**0505**] Once all software components are verified, the master SMS opens (and decompresses, if necessary) an upgrade instruction file also included as one of the software components loaded into sub-directory **1220** from the Installation Kit. The upgrade instruction file indicates the scope of the upgrade (i.e., upgrade mode). For instance, the upgrade instruction file may indicate that the upgrade may be hot or cold or must only be cold. The upgrade instruction file may also indicate that the upgrade may be done only across the entire chassis—that is, all applications to be upgraded must be upgraded simultaneously across the entire chassis—or that the upgrade may be done on a board-by-board basis or a path-by-path basis or some other partial chassis upgrade. A board-by-board upgrade may allow a network device administrator to chose certain boards on which to upgrade applications and allow older versions of the same applications to continue running on other boards. Similarly, path-by-path or other service related upgrades may allow the network administrator to chose to upgrade only the applications controlling particular services for particular customers, for example, a single path, while allowing older versions of the applications to continue to control the other services. Various upgrade modes are possible.

[**0506**] The upgrade instructions file may also include more detailed instructions such as the order in which each software component should be upgraded. That is, if several applications are to be upgraded, certain ones may need to be upgraded before certain other ones. Similarly, certain software components may need to be upgraded simultaneously. Moreover, certain boards may need to be upgraded prior to other boards. For example, control processor card **12** may need to be upgraded prior to upgrading any line cards.

Nothing is done to monitor the board's (compare to Applicants "embedded device") attributes; all actions are taken by the Master SMS. Paragraph [0504] does do polling, but it polls the installation directory, not the embedded device.

C. The Examiner's Argument

In the most recent office action, the Examiner responds to the applicant's arguments with

> The examiner maintains that the availability of upgrades, along with the board specific upgrade instructions (paragraphs [0505] and [0506]) may be considered attributes of the embedded device in accordance with the *monitoring program code* of claim 1. The master SMS

As such, the Examiner considers the addition of a download file into a directory on the SMS to be a change in the attributes of the embedded device.

D. Predetermined Attribute of Embedded Device Element Missing

The problem with the logic in the Final Office Action is that the attributes in Reynolds are attributes of the SMS server, and not attributes of the embedded device. The language in the claim, "...a change of at least one predetermined attribute of said embedded device...", clearly indicates that the change is occurring in the embedded device. The attribute is "of said embedded device" and not simply something associated with the embedded device.

Reynolds' download files are generic, and Reynolds describes that they may be downloaded to any of the devices. They are not specific to the embedded device, but are separate from the device and are changed independently of the embedded device. They are not attributes of the embedded device.

Furthermore, these files are not predetermined. The files in Reynolds arrive asynchronously and will be unique. There is nothing predetermined about the files to download.

Therefore, the Reynolds' files are not "...predetermined attributes of said embedded device..."

This element is simply missing from the teachings of Reynolds, and claims 1and 10 are not anticipated. The rejection under 35 U.S.C. § 102(e) can not be sustained, and the Applicants request that the Board reverse the Examiner and order that these claims be allowed.

E. Dependent Claims

Claims 2-9 and 11-17 depend upon claim 1 or 10 and are therefore distinct from Reynolds for the above reasons.

F. Claim 18 is patentable over the combination of Reynolds and Spring

Claim 18 stands rejected by the Examiner under 35 U.S.C. § 103(a) as being unpatentable over Reynolds in view of U.S. Patent No. 6,549,943, issued to Maximilian Spring *et al.* on April 15, 2003 (hereinafter, "Spring").

In the Final Office Action, the Examiner uses Spring to provide the computer program product elements of claim 18. Spring relates to "information that defines one or more network devices for use with a network management system", and has does not apply to the download of embedded devices.

As such, the Examiner looks to Reynolds to supply the remaining elements of claim 18. However, as described above, Reynolds does not teach the "at least one event indication in response to a change of at least one predetermined attribute of said embedded device" element of claim 18.

Since this element is missing from the combination of Reynolds and Spring, the rejection under 35 U.S.C. § 103(a) can not be sustained, and Applicants request that the Board reverse the Examiner and order that claim 18 be allowed.

VIII. CLAIMS APPENDIX

Listing of Claims:

1. (Previously Presented) A system for replacing a code image in an embedded device, comprising:

control program code responsive to at least one user command for issuing a plurality of device commands including at least one device command to replace said code image in said embedded device;

monitoring program code, asynchronous with respect to said control program code, for generating at least one event indication in response to a change of at least one predetermined attribute of said embedded device and forwarding said at least one event indication to said control program code; and

wherein said at least one device command replaces said code image in response to said at least one event indication.

2. (Original) The system of claim 1, wherein said control program code and said monitoring program code are independent threads of execution.

3. (Previously Presented) The system of claim 1, further comprising a device abstraction software object, wherein said device abstraction software object generates at least one event to said monitoring program code in response to information obtained from said embedded device.

4. (Previously Presented) The system of claim 3, wherein said device abstraction software object generates at least one event to said control program code in response to information obtained from said embedded device.

5. (Original) The system of claim 4, wherein said information obtained from said embedded device includes at least one value from a Management Information Base (MIB) stored on said embedded device.

6. (Previously Presented) The system of claim 3, wherein said device abstraction software object further operates to receive said at least one command from said control program code, and, in response to said at least one command from said control program code, send at least one corresponding query to said embedded device.

7. (Previously Presented) The system of claim 3, wherein said monitoring program code operates to periodically check the state of at least one attribute of said embedded device.

8. (Previously Presented) The system of claim 7, wherein said monitoring program code operates to periodically check said state of said at least one attribute of said embedded device by sending at least one command to said device abstraction software object

9. (Original) The system of claim 1, further comprising a state machine, wherein said state machine is represented in program code accessible to said control program code.

10. (Previously Presented) A method for replacing a code image in an embedded device, comprising:

issuing, responsive to at least one user command, a plurality of device commands including at least one device command to replace said code image in said embedded device, wherein said issuing is performed by control program code;

generating, asynchronous with respect to said control program code, at least one event indication in response to a change of at least one predetermined attribute of said embedded device and forwarding said at least one event indication to said control program code, wherein said generating is performed by monitoring program code; and

Wherein said at least one device command replaces said code image in said embedded device, and wherein said at least one device command is generated responsive to said at least one event indication.

11. (Previously Presented) The method of claim 10, wherein said at least one event is generated to said monitoring program code by a device abstraction software object, and wherein said generating of said at least one event by said device abstraction software object is in response to information obtained from said embedded device.

12. (Previously Presented) The method of claim 11, wherein said generating by said device abstraction software object of said at least one event to said control program code is responsive to obtaining information from said embedded device by said device abstraction software object.

13. (Previously Presented) The method of claim 12, wherein said obtaining information from said embedded device includes obtaining at least one value from a Management Information Base (MIB) stored on said embedded device.

14. (Previously Presented) The method of claim 13, further comprising receiving, by said device abstraction software object, said at least one command from said control program code, and, in response to said at least one command from said control program code, sending at least one corresponding query to said embedded device.

15. (Previously Presented) The method of claim 11, further comprising periodically checking, by said monitoring program code, the state of at least one attribute of said embedded device.

16. (Previously Presented) The method of claim 15, further comprising, periodically checking, by said monitoring program code, said state of said at least one attribute of said embedded device by sending at least one command to said device abstraction software object.

17. (Original) The method of claim 10, further comprising maintaining a current state of said embedded device in a state machine, wherein said state machine is represented in a program code accessible to said program code.

18. (Previously Presented) A computer program product including a computer readable medium, said computer readable medium having a computer program stored thereon, said computer program for upgrading a software image on an embedded device, said computer program comprising:

control program code for issuing, responsive to at least one user command, a plurality of device commands including at least one device command to replace said code image in said embedded device;

monitoring program code for generating asynchronous with respect to said control program code, at least one event indication in response to a change of at least one predetermined attribute of said embedded device and forwarding said at least one event indication to said control program code; and

wherein said at least one device command replaces said code image in said embedded device, and wherein said at least one device command is generated responsive to said at least one event indication.

19. (Previously Presented) A system for upgrading a software image on an embedded device, said computer program comprising:

means for controlling an upgrade process, said means for controlling including means for issuing, responsive to at least one user command, a plurality of device commands including at least one device command to replace said code image in said embedded device;

means for monitoring an embedded device, wherein said means for monitoring includes means for generating, asynchronous with respect to said means for controlling, at least one event indication in response to a change of at least one predetermined attribute of said embedded device and forwarding said at least one event indication to said control program code; and

wherein said at least one device command replaces said code image in said embedded device, and wherein said at least one device command is generated responsive to said at least one event indication.

20. (Currently Amended) A system for replacing a code image in an embedded device, comprising:

a control program operative, responsive to a user command, to replace said code image in said embedded device; and

a monitor program operative, asynchronously with respect to said control program, to:

monitor progress of replacing said code image in said embedded device; and

generate an event indication to said control program to indicate a status of replacing said code image after replacement of said code image has begun but before replacement of said code image is completed.

21. (Previously Presented) The system of claim 20, wherein said monitor program is further operative to:

detect a failure during said replacement of said code image; and

generate said event indication to said control program in response to detecting said failure.

22. (Previously Presented) The system of claim 20, wherein said monitor program is further operative to:

monitor a number of bytes received by said embedded device during replacement of said code image; and

generate said event indication to said control program in response to monitoring said number of bytes received by said embedded device.

23. (Previously Presented) The system of claim 20, wherein said monitor program is further operative to:

monitor a number of files received by said embedded device during replacement of said code image; and

generate said event indication to said control program in response to monitoring said number of files received by said embedded device.

24. (Previously Presented) The system of claim 20, wherein said monitor program is further operative to:

monitor said embedded device for a reset operation performed by said embedded device; and

generate said event indication to said control program in response to said reset operation performed by said embedded device.

25. (Previously Presented) The system of claim 20, wherein said control program and said monitoring program are independent threads of execution.

26. (Previously Presented) The system of claim 20, further comprising a device abstraction software object operative to generate at least one event to said monitor program in response to information obtained from said embedded device.

27. (Previously Presented) The system of claim 26, wherein said device abstraction software object is further operative to generate at least one event to said control program in response to information obtained from said embedded device.

28. (Previously Presented) The system of claim 27, wherein said information obtained from said embedded device includes at least one value from Management Information Base (MIB) stored on said embedded device.

29 (Previously Presented) The system of claim 26, wherein said device abstraction software object is further operative, in response to receiving a command from said control program, to send at least one corresponding query to said embedded device.

30. (Previously Presented) The system of claim 26, wherein said monitor program is further operative to periodically check the state of at least one attribute of said embedded device.

31. (Previously Presented) The system of claim 30, wherein said monitor program code is further operative to periodically check said state of said at least one attribute of said embedded device by sending at least one command to said device abstraction software object.

32. (Previously Presented) The system of claim 20, further comprising a state machine represented in program code accessible to said control program.

33. (Previously Presented) A method for replacing a code image in an embedded device, comprising

responsive to a user command, replacing said code image in said embedded device;

asynchronously, with respect to replacing said code image, monitoring progress of replacing said code image in said embedded device; and

generating an event indication to indicate a status of replacing said code image after replacement of said code image has begun but before replacement of said code image is completed.

34. (Previously Presented) The method of claim 33 further comprising:

detecting a failure during said replacement of said code image; and

wherein said generating said event indication comprises generating said event indication in response to detecting said failure.

35. (Previously Presented) The method of claim 33, further comprising:

monitoring a number of bytes received by said embedded device during replacement of said code image; and

wherein said generating said event indication comprises generating said event indication in response to monitoring said number of bytes received by said embedded device.

36. (Previously Presented) The method of claim 34, further comprising:

monitoring a number of files received by said embedded device during replacement of said code image; and

wherein said generating said event indication comprises generating said event indication in response to monitoring said number of files received by said embedded device.

37. (Currently Amended) The method of claim 33, further comprising:

monitoring said embedded device for a reset operation performed by said embedded devices; and

wherein said generating said event indication comprises generating said event indication in response to said reset operation preformed by said embedded device.

IX. EVIDENCE APPENDIX

None

X. RELATED PROCEEDINGS APPENDIX

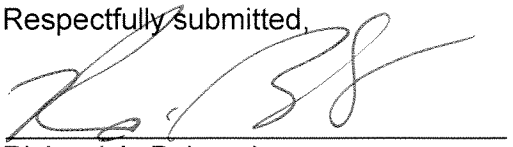
None

CONCLUSION

The pending claims define subject matter that is distinct from Reynolds both independently and in combination with either Ording or Spring. Therefore the pending claims are patentable under 35 U.S.C. § 102(e) and 35 U.S.C. § 103(a). Claims 1-18 are pending and in condition for allowance.

Applicants respectfully request that the Board reverse the outstanding rejections and direct the Examiner to promptly issue this application.

Respectfully submitted,



Richard A. Baker, Jr.
Registration No. 48, 124
3COM CORPORATION
350 Campus Drive
Marlborough, MA 01752
Telephone: 508-323-1085